Understanding and Describing Bayesian Models with bayestestR

09 June 2019

Introduction

The Bayesian framework for statistics is quickly gaining in popularity among scientists, for reasons such as reliability and accuracy (particularly in noisy data and small samples), the possibility to incorporate prior knowledge into the analysis or the intuitive interpretation of results (Andrews and Baguley 2013; Etz and Vandekerckhove 2016; Kruschke 2010; Kruschke, Aguinis, and Joo 2012; Wagenmakers et al. 2018). Adopting the Bayesian framework is more of a shift in the paradigm than a change in the methodology; All the common statistical procedures (t-tests, correlations, ANOVAs, regressions, etc.) can also be achieved within the Bayesian framework. One of the core difference is that in the *frequentist* view, the effects are fixed (but unknown) and data are random. On the other hand, in the Bayesian inference process, instead of having estimates of the "true effect", the probability of different effects *given the observed data* is computed, resulting in a distribution of possible values for the parameters, called the *posterior distribution*.

The uncertainty in Bayesian inference can be summarized, for instance, by the *median* of the distribution, as well as a range of values of the posterior distribution that includes the 95% most probable values (the 95% *credible interval*). Cum grano salis, these are considered the counterparts to the point-estimate and confidence interval in a frequentist framework. To illustrate the difference of interpretation, the Bayesian framework allows to say *"given the observed data, the effect has 95% probability of falling within this range"*, while the frequentist less straightforward alternative would be *"when repeatedly computing confidence intervals from data of this sort, there is a 95% probability that the effect falls within a given range"*. In essence, the Bayesian sampling algorithms (such as MCMC sampling) return a probability distribution (*the posterior*) of an effect that is compatible with the observed data. Thus, an effect can be described by characterizing its posterior distribution in relation to its centrality (point-estimates), uncertainty, as well as existence and significance (difference from a null). We have to point out that, although we use the term *null-hypothesis significance testing* (NHST) in this paper, which is possible to carry out in a Bayesian framework, "Bayesian NHST" is usually extended to general testing of "effect existence" - not just of a null-hypothesis - and thus does not perfectly map onto statistical significance testing in its classical meaning, with associated p-values. This should become clearer in the following sections.

Existing R packages allow users to easily fit a large variety of models and extract and visualize the posterior draws. However, most of these packages only return a limited set of indices (*e.g.*, point-estimates and CIs). `bayestestR` provides a comprehensive and consistent set of functions to analyze and describe posterior distributions generated by a variety of models objects, including popular modeling packages such as `rstanarm` (Goodrich et al. 2018), `brms` (Bürkner and others 2017) or `BayesFactor` (Morey and Rouder 2018). Beyond computing point-estimates (mean, median or MAP estimate) and quantifying the related uncertainty (with different types of CIs), `bayestestR` focuses on implementing a Bayesian null-hypothesis testing framework. By providing access to both established and exploratory indices of effect existence and significance, `bayestestR` appears as a useful tool supporting Bayesian statistics. The main functions are described below, and a full documentation is available on the package's website.

Examples of Features

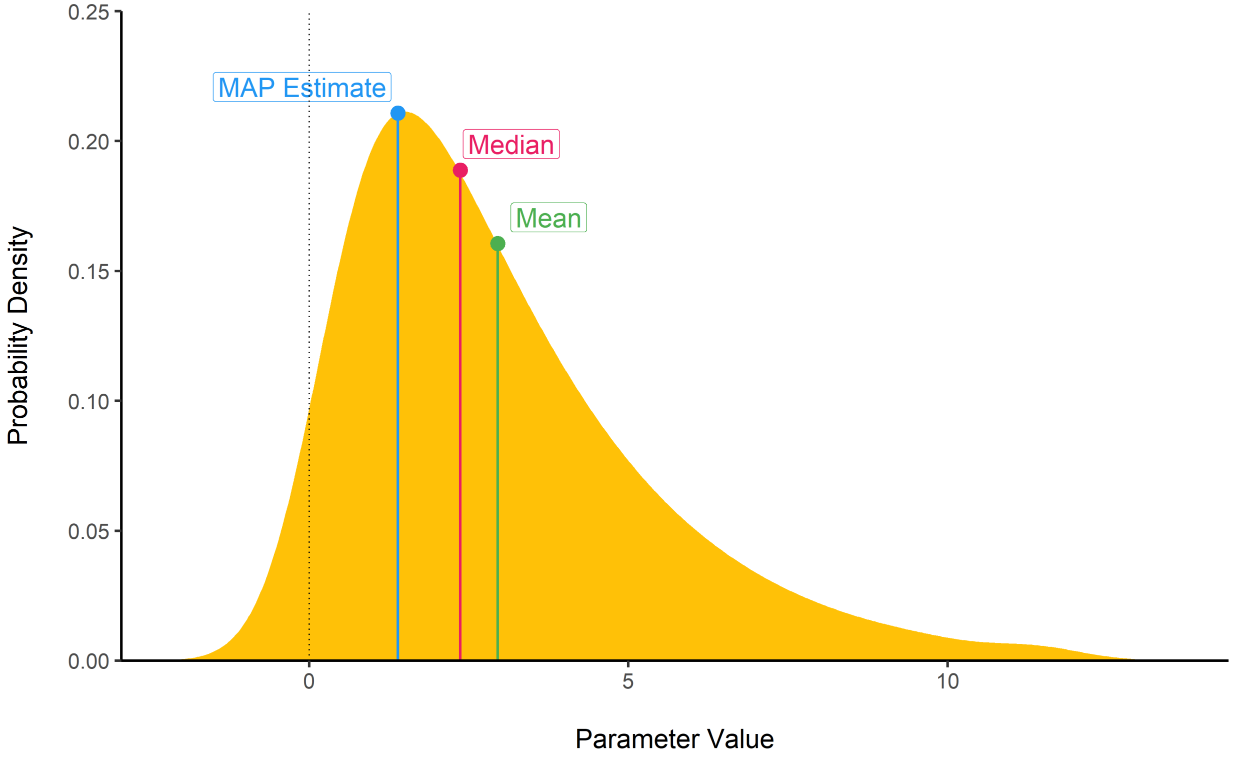
The following demonstration of functions in `bayestestR` is accompanied by figures. However, these figures are not produced by the `bayestestR` functions, but rather serve to illustrate the conceptional ideas behind the related indices. Plotting capability for the functions in `bayestestR` are available in the `see`-package.

Indices of Centrality: Point-estimates

bayestestR offers two functions to compute point-estimates from posterior distributions: map_estimate() and point_estimate(), the latter providing options to calculate the mean, median or MAP estimate of a posterior distribution. map_estimate() is a convenient function to calculate the MAP directly.

The **posterior mean** minimizes expected *squared* error, whereas the **posterior median** minimizes expected *absolute* error (i.e. the difference of estimates from true values over samples). The highest **Maximum A Posteriori** (MAP) estimate is the most probable value of a posterior distribution.

```
set.seed(1)
posterior <- rchisq(100, 3)
map_estimate(posterior)
#> MAP = 1.46

point_estimate(posterior)
#> Median = 2.31

point_estimate(posterior, centrality = "mean")
#> Mean = 2.96

point_estimate(posterior, centrality = "map")
#> MAP = 1.46
```



Indices of Uncertainty: HDI and CI

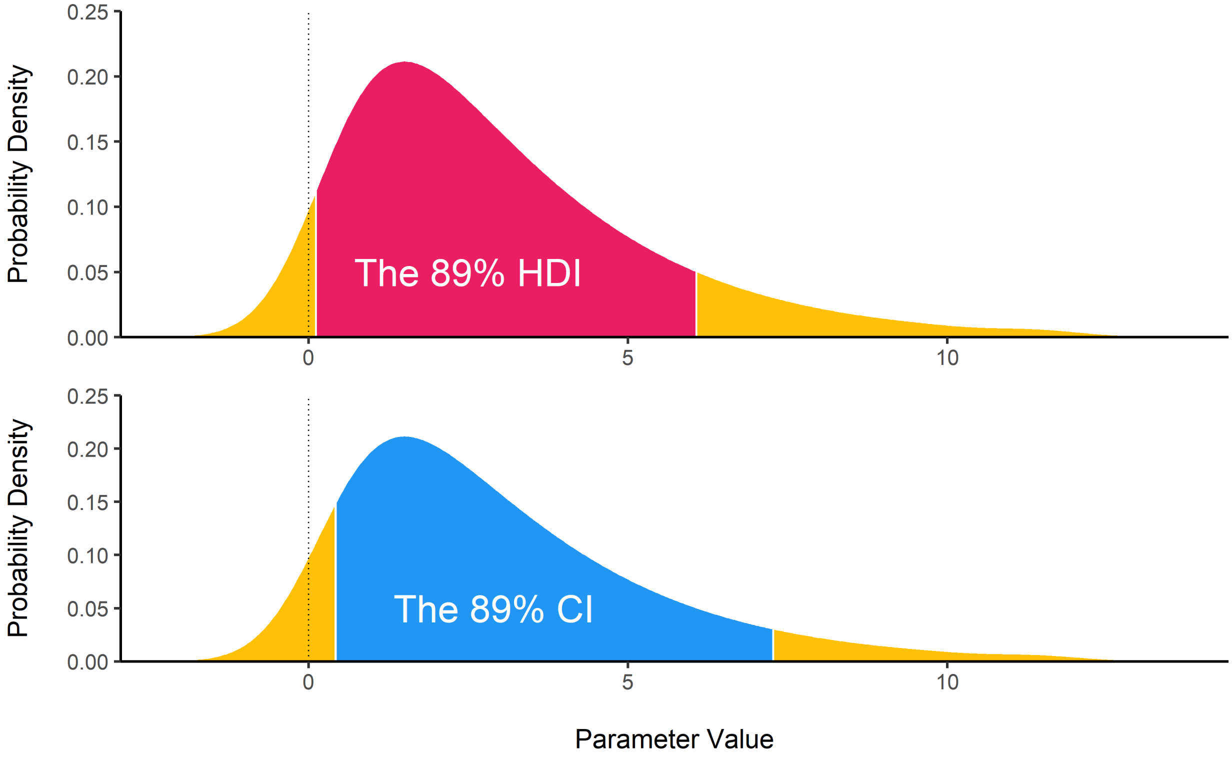
To measure the uncertainty in the estimation, bayestestR provides two functions: ci(), the “classical”, equal-tailed credible interval, and hdi(), the highest density interval.

hdi() computes the **Highest Density Interval (HDI)** of a posterior distribution, i.e., the interval which contains all points within the interval have a higher probability density than points outside the interval. The

HDI can be used in the context of Bayesian posterior characterisation as **Credible Interval (CI)**, however, HDIs have a particular property: Unlike equal-tailed intervals (see `ci()`) that typically exclude 2.5% from each tail of the distribution, the HDI is *not* equal-tailed and therefore always includes the mode(s) of posterior distributions.

By default, `hdi()` and `ci()` return the 89% intervals (`ci = 0.89`), deemed to be more stable than, for instance, 95% intervals. An effective sample size of at least 10.000 is recommended if 95% intervals should be computed (Kruschke 2015). Moreover, 89 is the highest prime number that does not exceed the already unstable 95% threshold (McElreath 2018).

```
hdi(posterior)
#> # Highest Density Interval
#>
#>       89% HDI
#>  [0.11, 6.05]

ci(posterior)
#> # Credible Interval
#>
#>        89% CI
#>  [0.42, 7.27]
```



Null-Hypothesis Significance Testing (NHST)

ROPE

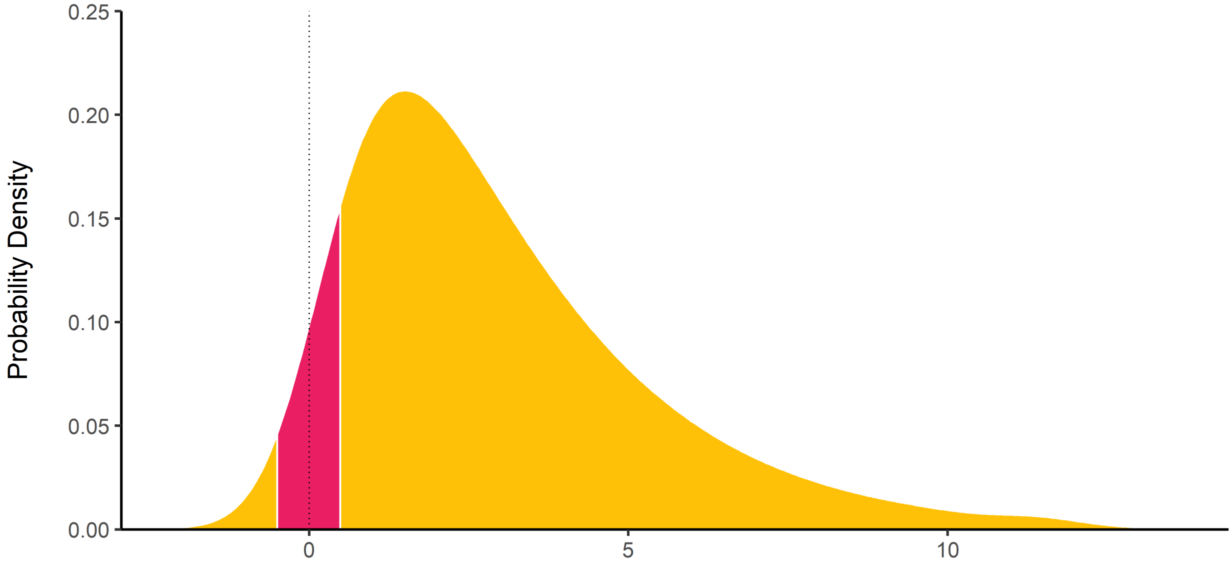
`rope()` computes the proportion (in percentage) of the HDI (default to the 89% HDI) of a posterior distribution that lies within a region of practical equivalence.

Statistically, the probability of a posterior distribution of being different from 0 does not make much sense (the probability of it being different from a single point being infinite). Therefore, the idea underlining ROPE is to let the user define an area around the null value enclosing values that are *equivalent to the null* value for practical purposes (John K Kruschke and Liddell 2018; John K. Kruschke 2018).

Kruschke suggests that such null value could be set, by default, to the -0.1 to 0.1 range of a standardized parameter (negligible effect size according to Cohen, 1988). This could be generalized: For instance, for linear models, the ROPE could be set as `0 +/- .1 * sd(y)`. This ROPE range can be automatically computed for models using the rope_range function.

The proportion of the 95% HDI (or 90%, considered more stable) that falls within the ROPE can be used as an index for "null-hypothesis" testing (as understood under the Bayesian framework, see equivalence_test).

```
rope(posterior, range = c(-0.5, 0.5))
#> # Proportion of samples inside the ROPE [-0.50, 0.50]:
#> 
#>  inside ROPE
#>       8.89 %
```



Test for Practical Equivalence

`equivalence_test()` is a **Test for Practical Equivalence** based on the "HDI+ROPE decision rule" (John K. Kruschke 2018) to check whether parameter values should be accepted or rejected against an explicitly formulated "null hypothesis" (i.e., a ROPE).

The percentage of the HDI that falls within the ROPE serves as decision rule: If the HDI is completely outside the ROPE, the "null hypothesis" for this parameter is "rejected". If the ROPE completely covers the HDI, i.e., all most credible values of a parameter are inside the region of practical equivalence, the null hypothesis is accepted. Else, it's undecided whether to accept or reject the null hypothesis.

```
equivalence_test(posterior, range = c(-0.5, 0.5))
#> # Test for Practical Equivalence
#> 
#>   ROPE: [-0.50 0.50]
#> 
#>         H0 inside ROPE     89% HDI
#>  Undecided      8.89 % [0.11 6.05]
```

As said above, for regression models `equivalence_test()` will automatically find an appropriate range for the ROPE. However, it is also possible to define a custom range using the `range`-argument.

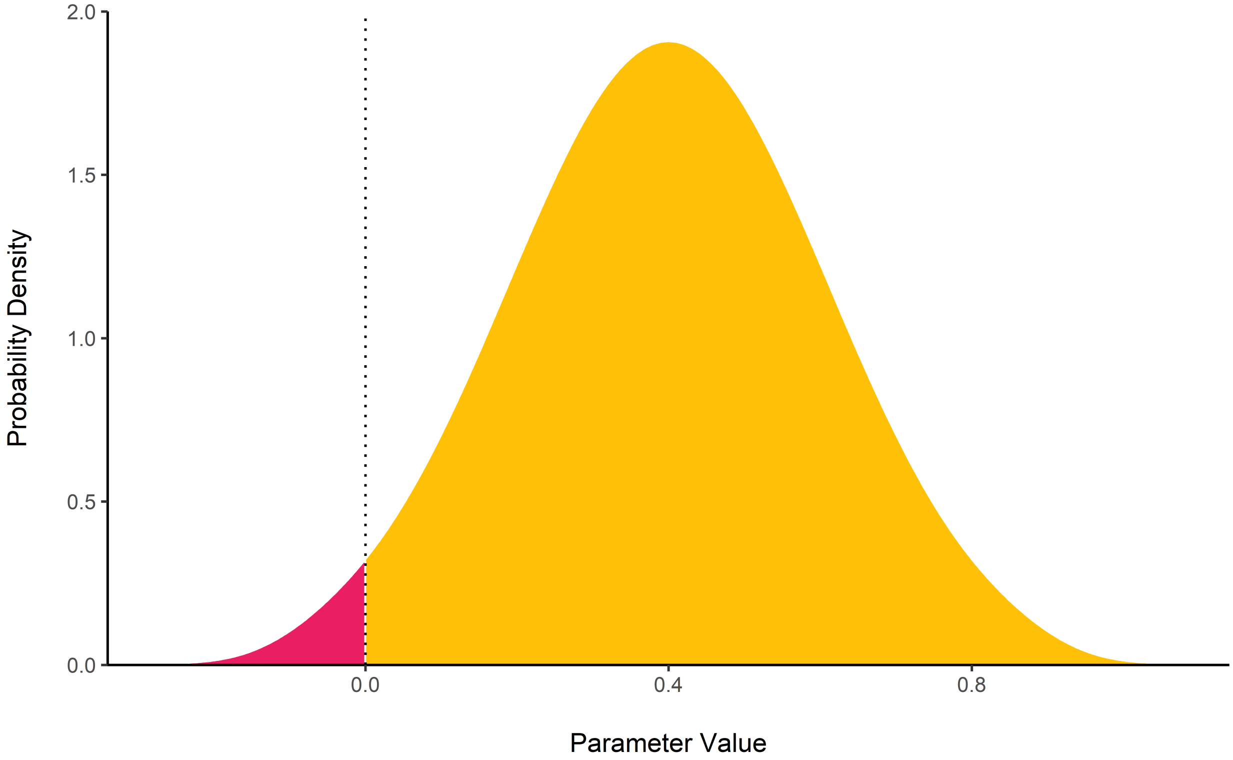
```
library(rstanarm)
model <- stan_glm(mpg ~ wt + gear, data = mtcars)
equivalence_test(model)
#> # Test for Practical Equivalence
#>
#>   ROPE: [-0.60 0.60]
#>
#>    Parameter        H0 inside ROPE       89% HDI
#>  (Intercept)  Rejected      0.00 % [30.82 47.02]
#>           wt  Rejected      0.00 % [-6.63 -4.39]
#>         gear Undecided     52.54 % [-1.76  1.23]
```

Probability of Direction (*pd*)

`p_direction()` computes the **Probability of Direction** (*pd*, also known as the Maximum Probability of Effect - *MPE*). It varies between 50% and 100% (i.e., `0.5` and `1`) and can be interpreted as the probability (expressed in percentage) that a parameter (described by its posterior distribution) is strictly positive or negative (whichever is the most probable). It is mathematically defined as the proportion of the posterior distribution that is of the median's sign. Although differently expressed, this index is fairly similar (i.e., is strongly correlated) to the frequentist *p-value*: the *pd* corresponds to the frequentist one-sided p-value through the formula `p-value = (1-pd/100)` and to the two-sided p-value (the most commonly reported) through the formula `p-value = 2*(1-pd/100)`. Thus, a `pd` of `95%`, `97.5%` `99.5%` and `99.95%` corresponds approximately to a two-sided *p*-value of respectively `.1`, `.05`, `.01` and `.001`. See the *reporting guidelines*.

The demonstration of the *pd* only makes sense for distribution that in principle can have both positive and negative values, so we use a normal distribution with values approximately ranging from `-.1` to `.9` for the next example.

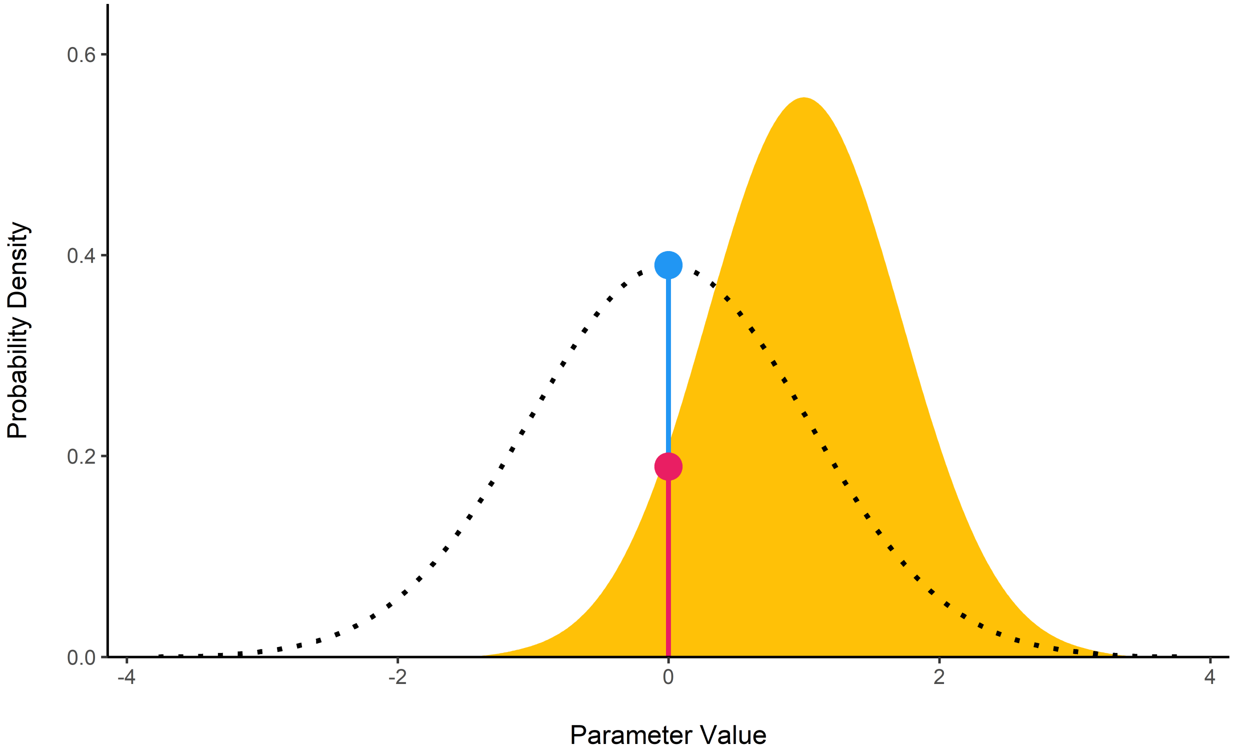
```
p_direction(distribution_normal(100, 0.4, 0.2))
#> # Probability of Direction (pd)
#>
#> pd = 98.00%
```



Bayes Factor

`bayesfactor_savagedickey()` computes the ratio between the density of a single value (typically the null) in two distributions. When these distributions are the prior and the posterior distributions, this ratio can be used to examine the degree by which the mass of the posterior distribution has shifted further away from or closer to the null value (relative to the prior distribution), thus indicating if the null value has become less or more likely given the observed data. The Savage-Dickey density ratio is also an approximation of a Bayes factor comparing the marginal likelihoods of the model against a model in which the tested parameter has been restricted to the point null (Wagenmakers et al. 2010).

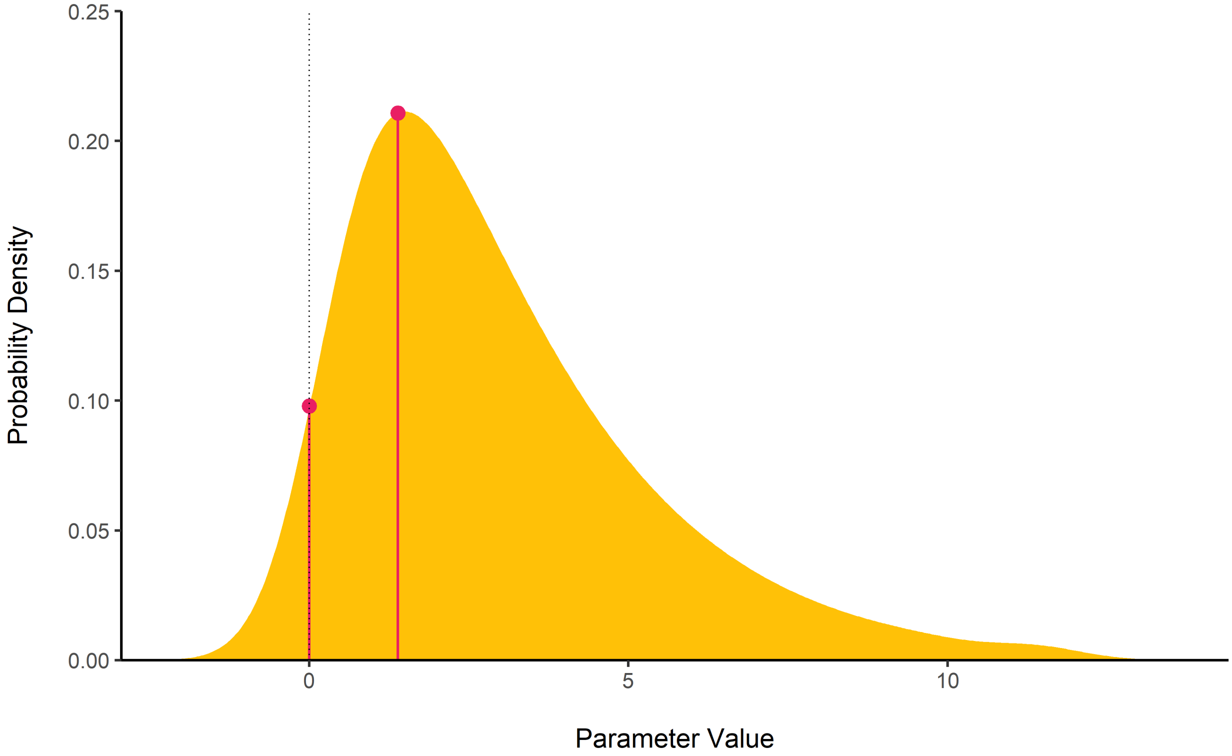
```r
prior <- rnorm(1000, mean = 0, sd = 1)
posterior <- rnorm(1000, mean = 1, sd = 0.7)

bayesfactor_savagedickey(posterior, prior, direction = "two-sided", hypothesis = 0)
```



MAP-based *p*-value

`p_map()` computes a Bayesian equivalent of the p-value, related to the odds that a parameter (described by its posterior distribution) has against the null hypothesis (*h0*) using Mills' (2014, 2017) *Objective Bayesian Hypothesis Testing* framework. It is mathematically based on the density at the Maximum A Priori (MAP) and corresponds to the density value at 0 divided by the density of the MAP estimate.

```
p_map(posterior)
#> # MAP-based p-value
#>
#> p (MAP) = 0.000
```



Licensing and Availability

bayestestR is licensed under the GNU General Public License (v3.0), with all source code stored at GitHub (https://github.com/easystats/bayestestR), with a corresponding issue tracker for bug-reporting and feature enhancements. In the spirit of open science and research, we encourage interaction with our package through requests/tips for fixes, feature updates, as well as general questions and concerns via direct interaction with contributors and developers.

Acknowledgments

We would like to thank the council of masters of easystats, all other padawan contributors as well as the users.

References

Andrews, Mark, and Thom Baguley. 2013. "Prior Approval: The Growth of Bayesian Methods in Psychology." *British Journal of Mathematical and Statistical Psychology* 66 (1): 1–7.

Bürkner, Paul-Christian, and others. 2017. "Brms: An R Package for Bayesian Multilevel Models Using Stan." *Journal of Statistical Software* 80 (1): 1–28.

Etz, Alexander, and Joachim Vandekerckhove. 2016. "A Bayesian Perspective on the Reproducibility Project: Psychology." *PloS One* 11 (2): e0149794.

Goodrich, Ben, Jonah Gabry, Imad Ali, and Sam Brilleman. 2018. "Rstanarm: Bayesian Applied Regression Modeling via Stan." http://mc-stan.org/.

Kruschke, John K. 2010. "What to Believe: Bayesian Methods for Data Analysis." *Trends in Cognitive Sciences* 14 (7): 293–300.

Kruschke, John K. 2015. *Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and Stan.* 2. ed. Amsterdam: Academic Press.

———. 2018. “Rejecting or Accepting Parameter Values in Bayesian Estimation.” *Advances in Methods and Practices in Psychological Science*, 251524591877130. https://doi.org/10.1177/2515245918771304.

Kruschke, John K, Herman Aguinis, and Harry Joo. 2012. “The Time Has Come: Bayesian Methods for Data Analysis in the Organizational Sciences.” *Organizational Research Methods* 15 (4): 722–52.

Kruschke, John K, and Torrin M Liddell. 2018. “The Bayesian New Statistics: Hypothesis Testing, Estimation, Meta-Analysis, and Power Analysis from a Bayesian Perspective.” *Psychonomic Bulletin & Review* 25 (1): 178–206.

McElreath, Richard. 2018. *Statistical Rethinking: A Bayesian Course with Examples in R and Stan.* Chapman; Hall/CRC.

Morey, Richard D., and Jeffrey N. Rouder. 2018. *BayesFactor: Computation of Bayes Factors for Common Designs.* https://CRAN.R-project.org/package=BayesFactor.

Wagenmakers, Eric-Jan, Tom Lodewyckx, Himanshu Kuriyal, and Raoul Grasman. 2010. “Bayesian Hypothesis Testing for Psychologists: A Tutorial on the Savage–Dickey Method.” *Cognitive Psychology* 60 (3): 158–89.

Wagenmakers, Eric-Jan, Maarten Marsman, Tahira Jamil, Alexander Ly, Josine Verhagen, Jonathon Love, Ravi Selker, et al. 2018. “Bayesian Inference for Psychology. Part I: Theoretical Advantages and Practical Ramifications.” *Psychonomic Bulletin & Review* 25 (1): 35–57.